

УДК 004.9

РАЗРАБОТКА АЛГОРИТМА ПОСТРОЕНИЯ ДЕРЕВА ВЫИГРЫШНОЙ СТРАТЕГИИ ИГРЫ С ПОЛНОЙ ИНФОРМАЦИЕЙ

Шумская Е. А.

Самарский национальный исследовательский университет
имени академика С. П. Королёва, г. Самара

ЕГЭ по информатике состоит из двух частей. Первая часть содержит задания с кратким ответом, вторая - с развернутым. Задания второй части проверяются экспертами на соответствие особым критериям. Одним из творческих заданий является задача «о камнях». На настоящий момент практически не существует программ, которые помогали бы выпускникам проверять решения заданий данного типа, при этом они предоставляют только ответ без какого-либо пояснения решения. Это усложняет подготовку учащихся к экзамену и оценку собственных способностей. В связи с этим становится актуальной задача разработки алгоритма автоматического построения дерева для определения выигрышной стратегии и реализации на его основе модуля, интегрированного в среду LMS Moodle. С его помощью преподаватель сможет составлять задачи «о камнях» и сохранять их в базу данных (БД), хранящуюся на удаленном сервере, а обучаемый сможет решать задания «о камнях» в режиме online, модуль обеспечит динамическую визуализацию процесса решения и автоматическую проверку решения с учетом критериев оценки.

Задачу о камнях в общем виде можно сформулировать следующим образом. У двух игроков есть n куч камней, в каждой из которых k камней. Задан набор возможных операций над камнями из одной кучи за один ход (например, добавить один камень в кучу, увеличить количество камней в два раза) и условие завершения игры. Для того чтобы делать ходы, у каждого игрока есть неограниченное количество камней.

Задача о камнях является комбинаторной беспристрастной нормальной игрой и обладает всеми её основными свойствами:

- игроки ходят по очереди;
- оба игрока знают все ходы, сделанные в игре;
- игра заканчивается, когда достигнута позиция, в которой нет возможных ходов для игрока, чья очередь ходить, – во всех таких позиция определено, кто из игроков становится победителем;
- выигрывает игрок, сделавший последний ход.

Игрок имеет выигрышную стратегию, если он может выиграть при любых ходах противника. Описать стратегию игрока – значит описать, какой ход он должен сделать в любой ситуации, которая ему может встретиться при различной игре противника. Решением задачи является дерево выигрышной стратегии в виде графа [1].

Сначала строится дерево игры, включающее в себя все возможные ходы игроков. Затем ведется поиск победителя. И в итоге, зная победителя, можно отсечь лишние ветки решения. Полученное дерево и будет представлять собой эталонное решение.

Первая часть алгоритма – построение дерева игры. Пример построения дерева игры для двух игроков приведён в работе [2]. Сначала создается корневая вершина/состояние, которая описывает исходную ситуацию – исходное количество камней во всех кучах. Поместив ее в очередь, запускается рекурсивная процедура, состоящая из следующих шагов:

- 1) извлечение текущего состояния из очереди,

2) проверка на то, является ли оно заключительным, то есть сумма всех камней в кучах превышает или равна определенному значению.

3) если нет, то создаются все возможные состояния, которые может получить следующий игрок, они хранятся как потомки текущего состояния.

4) если среди них возникли заключительные состояния, будут оставлены только они, иначе сохраняются все; сохраненные состояния помещаются в очередь.

5) в случае, если текущее состояние было выигрышным, происходит выход, при котором не создаются потомки для такого состояния.

Вторая часть основного алгоритма – поиск победителя. Победитель или обладатель выигрышной стратегии – это такой игрок, который выигрывает при любом ходе противника. Процедура поиска следующая: прежде всего каждое из возможных состояний помечается выигрышным или проигрышным. Выигрышное состояние – заключительное или такое, что из него тот же игрок может получить заключительное состояние. Для этого запускается рекурсивная процедура из всех состояний, которые может получить первый игрок, просматриваются все следующие за ним состояния, которые может получить второй игрок после хода первого игрока. Если среди них есть хотя бы одно выигрышное, то исходное состояние первого игрока помечается проигрышным, иначе, если состояние проигрышное увеличиваем счетчик, если же состояние еще не было помечено, запускается данная рекурсивная процедура и оно помечается. Снова проверяется его значение и в зависимости от него корневое состояние помечается выигрышным или проигрышным. После завершения просмотра всех следующих возможных состояний (потомков) из текущего проверяется значение счетчика, если оно равно их общему числу, корневое состояние помечается выигрышным, иначе – проигрышным. После выхода из рекурсивной процедуры производится проверка значений всех возможных состояний первого игрока при совершении им своего первого хода. Если все они отмечены как выигрышные, результатом будет победа первого игрока, иначе – второго.

Третья часть алгоритма – отсеечение лишних веток необходима для отбрасывания веток, хранящих информацию о состояниях/ходах выигравшего игрока, ведущих к выигрышу его оппонента, которые не могли произойти, если бы первый сделал другой ход из числа возможных.

Разработанный алгоритм используется в модуле, интегрированном систему дистанционного обучения Школы информатики СГАУ, построенной на базе LMS Moodle, и будет использоваться школьниками при подготовке ЕГЭ по информатике.

Библиографический список

1. Шумская Е. А., Зеленко Л. С. Решение проблемы хранения данных при разработке программного модуля для LMS Moodle // Международный научный журнал «Современные информационные технологии и ИТ-образование». Т.12. № 3-1. [под ред. В.А. Сухомлина]. М.: Изд-во «Фонд содействия развитию интернет-медиа, ИТ-образования, человеческого потенциала «Лига интернет-медиа», 2016. С. 97-102.
2. Зеленко Л. С., Шумская Е. А., Вилков А. В. Алгоритм построения оптимального дерева игры // Перспективные информационные технологии (ПИТ-2017): сб. науч. тр. межд. научно-техн. конф.; [под ред. С.А. Прохорова]. Самара: Изд-во СНЦ РАН, 2017. С.1070-1073.